Composer Help

Web Request Common Block

Web Request Common Block

Contents

The Web Request block is used for both routing and voice applications. Use to invoke any supported HTTP web request or REST-style web Service.

- It supports PUT, DELETE, GET and POST methods over HTTPS.

- It is based on common Web Services standards such as XML, SOAP and WSDL instead of proprietary standards that are currently being replaced.

REpresentational State Transfer (REST) is an XML-based protocol for invoking Web Services over HTTP. REST is a lighter version of SOAP, which has evolved into a more complex protocol. REST-style web services offer a less coupled paradigm whereby simpler requests and responses are used. As an example, a simple HTTP request follows the REST methodology. The Web Request block allows the user to query "RESTful" Web services. The supported return formats for the Web Request block are:

- plain text

Note: For workflows, the result will be returned in a JSON string with key name result, e.g., {"result":"This is a plain text result"}

- plain XML
- JSON string (See an issue pertaining to JSON objects in Troubleshooting.)

Important

Composer does not support fetching URLs using HTTPS in Web Request and Web Service blocks.

The Web Request block has the following properties:

Name Property

Find this property's details under Common Properties for Workflow Blocks or Common Properties for Callflow Blocks

Block Notes Property

Find this property's details under Common Properties for Workflow Blocks or Common Properties for Callflow Blocks.

Exceptions Property

Find this property's details under Common Properties for Workflow Blocks or Common Properties for Callflow Blocks You can also define custom events.

Request Method Property

This property Indicates the method for invoking the web request:

- **get**--Invoked using HTTP Get
- **post**--Invoked using HTTP Post. This option is valid only when the parameters are passed as a namelist (Use Namelist property is set to true). This is generally used when a large amount of data needs to be sent as an input value for a subdialog.
- **put**--Invoked using HTTP Put
- **delete**--Invoked using HTTP Delete

To select a value for the Request Method property:

1. Select the **Request Method** row in the block's property table.
2. In the **Value** field, select get, post, put, or delete from the drop-down list.

Uri Property

The Uri property specifies the http:// page to invoke. To set a URL destination for the Uri property:

1. Select the **Uri** row in the block's property table.
2. In the **Value** field, click the down arrow and select the variable that contains URL.

Condition Property

Find this property's details under Common Properties for Callflow Blocks or Common Properties for Workflow Blocks.

Logging Details Property

Find this property's details under Common Properties for Callflow Blocks or Common Properties for Workflow Blocks.

Log Level Property

Find this property's details under Common Properties for Callflow Blocks or Common Properties for Workflow Blocks.

Enable Status Property

Find this property's details under Common Properties for Callflow Blocks or Common Properties for Workflow Blocks.

Authentication Type Property

The Authentication Type property specifies whether to use an anonymous or basic authentication for the web request. To assign a value to the Authentication Type property:

1. Select the Authentication Type row in the block's property table.

2. In the Value field, select anonymous (default) or basic from the drop-down list. With the anonymous type of access, no user name/password is passed to Web service for client authentication in order to get data. If you select the basic type of access, you must supply the Login Name and Password properties.

Encoding Type Property

The Encoding Type property (used for callflows only) indicates the media encoding type of the submitted document. GVP 8.1 supports two encoding types:

- application/x-www-form-urlencoded

- multipart/form-data

To select a value for the Encoding Type property:

1. Select the **Encoding Type** row in the block's property table.

2. In the **Value** field, select one of the following:

    - **application/x-www-form-urlencoded (default)**

    - **application/json**

Input Parameters Property

Use the Input Parameters property to specify a list of required Name/Value pairs to pass as parameters to the http:// page. To specify input parameters:

1. Click the Parameters row in the block's property table.

2. Click the ⬛ button to open the Parameter Settings dialog box.

Add Button Use the Add button to enter parameter details.

1. Click **Add** to add an entry to Web Request Parameters.

2. In the **Parameter Name** field, accept the default name or change it.

3. From the **Parameter Type** drop-down list, select **In**, **Out**, or **InOut**:

In	Input parameters are variables submitted to the web request.
Out	Output parameters are variables that the web request returns and will be reassigned back to the current callflow/workflow.
InOut	InOut parameters are parameters that act as both input and output.

1. In the Expression drop-down list, select from among the variables shown, type your own expression, or click the ⬛ button to use Skill Expression Builder.

2. In the Definition field, type a description for this parameter.

3. Click Add again to enter another parameter, or click OK to finish.

Delete Button To delete a parameter:

1. Select an entry from the list.

2. Click **Delete**.

JSON Content Property

If the HTTP request to be invoked expects JSON content, this property can be used to specify that input. It expects a variable whose content will be sent to the API specified in the HTTP URI property of the block. Set the Encoding Type property of the block to application/json. In this case, the Input Parameters property will not be used.

The variable selected in this property should contain a JavaScript object. The object can be built from a JSON string, or using the ECMAScript block.

For example, if you would like to pass a JSON content to the HTTP URI, using a variable named "content", the variable can be initialized in the following ways:

- If you have a JSON string, you can use the Assign block to assign the following value to "content":

```
JSON.parse('{"abc": "def", "xyz": 3}')
```

- Alternately, you can build a JavaScript object using an ECMAScript block with code like the following:

```
var content = new Object(); content['abc'] = 'def'; content['xyz'] = 3;
```

In both cases, set the `JSON Content` property of the Web Request block to the variable named "content".

Timeout Property

Select the variable containing the number of seconds that the application will wait when fetching the result of the Web Service or the Web Request. If the requested resource does not respond in that time, then a timeout event will occur.

Custom HTTP Headers Property

Use this property to add Custom headers to be sent along with the HTTP request during the runtime execution of the Server Side block.

1. Click the row in the block's property table.
2. Click the ▨ button to open the Custom HTTP Headers dialog box.
3. Click **Add** to open Configuration Custom HTTP Headers dialog box.
4. Select a Header type.
5. Select **Literal** or **Variable**.
6. Type the literal value or select the variable that contains the value.

Login Name Property

Used when Authentication type = basic. The Login Name property specifies the login name for the invoked web page. To provide a login name for the web request:

1. Select the **Login Name** row in the block's property table.
2. In the **Value** field, type a valid login name.

Password Property

Used when Authentication type = basic. The Password property specifies the password for the invoked web page. To provide a password for the web request:

1. Select the **Password** row in the block's property table.

2. In the **Value** field, type a valid password that corresponds to the login name above.

Result Property

The Result property is the variable used to get back a result from the web request. To select a variable:

1. Select the **Result** row in the block's property table.

2. In the **Value** field, select one of the available variables from the drop-down list. Does not need to match the variable name that is coming back as a result of the web request.